Uma Arquitetura para Gerência de Identidades em Nuvens Híbridas

Guilherme Feliciano[1], Lucio Agostinho[1], Leonardo Olivi[1], Eliane G. Guimarães[2], Eleri Cardozo[1]

[1]Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas - UNICAMP

[2]Centro de Tecnologia da Informação Renato Archer
13083-970 - Campinas - SP

eleri@dca.fee.unicamp.br, eliane.guimaraes@cti.gov.br

Abstract. *Cloud computing offers a new paradigm to bring higher flexibility, higher reliability, and lower costs regarding the usage of computing resources. Although this approach leads to new business opportunities, the issue of security (specifically access control and identity management) is still an open and difficult to solve problem. This paper presents an architecture that provides federated identity management for hybrid clouds. The main contribution of this paper is a layered architecture for federated access control, where each layer is distributed in virtual machines. The OpenAM middleware was extended to include new services within the internal cloud network.*

Resumo. *A computação em nuvem oferece um novo paradigma para trazer maior flexibilidade, maior confiabilidade e menores custos com relação ao uso de recursos computacionais. Apesar desta abordagem levar a novas oportunidades de negócios, a questão da segurança (especificamente controle de acesso e gerência de identidades) ainda é um problema em aberto e de difícil solução. Este artigo apresenta uma arquitetura que contempla a gerência de identidades federadas em uma nuvem híbrida. A principal contribuição do trabalho é o controle de acesso federado, onde cada camada da arquitetura está distribuída em máquinas virtuais. O middleware OpenAM foi estendido para incluir novos serviços junto à rede interna da nuvem.*

1. Introdução

Na Internet de hoje é comum que os usuários tenham várias contas (identidades), que se traduzem em várias regras e diferentes permissões de acesso em cada um dos sites que visitam. Esta realidade acaba prejudicando a experiência de navegação do usuário, uma vez que o mesmo necessitará de várias contas nos diferentes sites. Com o surgimento da computação em nuvem os administradores de sistemas, por outro lado, têm agora que gerenciar estas contas sendo que, muitas vezes, estão replicadas para cada aplicação oferecida como serviço [Olden 2011]. Outro desafio é a natureza colaborativa que as aplicações em nuvem oferecem. Isto traz também problemas de segurança, uma vez que a nuvem precisa tratar questões como autenticação e autorização de identidades muitas vezes vindas de aplicações parceiras.

No setor acadêmico, ambientes de nuvem geralmente pertencem a domínios privados mantidos por centros de pesquisa nas universidades. Muitas vezes, nestes domínios, estão recursos cujo acesso pode ser realizado via Internet. Neste caso, é essencial que existam mecanismos para prover a autenticação e a autorização de usuários no acesso aos recursos, tanto para usuários vindos do próprio domínio, quanto de outras localidades. Esta questão possibilta uma interoperabilidade de recursos provindos de domínios distintos, habilitando então a cooperação entre centros de pesquisa parceiros.

Para que a cooperação seja realizada com êxito, as entidades parceiras devem definir políticas para o compartilhamento de recursos junto aos domínios federados. Esse processo envolve mecanismos de SSO (*Single Sign-On*) para que, uma vez que o usuário esteja autenticado em seu domínio de origem, ele possa acessar recursos em quaisquer outros domínios federados, desde que esse usuário esteja inserido no contexto de autorização do domínio em questão. Em domínios não-federados, o mesmo usuário precisaria ter credenciais em ambos os domínios para ter permissões de acesso aos recursos distribuídos [Cao and Yang 2010].

Este artigo apresenta como principal contribuição uma arquitetura para gerência de identidades distribuída com autenticação federada de identidades, autorização e SSO. Essa arquitetura é voltada para ambientes de computação em nuvem híbrida, notadamente para a integração de domínios privados e distintos que provêem acesso a recursos robóticos. O objetivo é apresentar um modelo para realizar o controle de acesso de domínios em nuvem, mas que estão inseridos em diferentes níveis de acesso: público e privado. Esse modelo é validado na plataforma REALcloud [Rocha et al. 2011] para promover SSO entre os diferentes sites de gerência nesta nuvem híbrida. O ambiente também oferece mecanismos para o acesso aos recursos da rede privada por meio de uma nuvem pública.

O artigo está organizado da seguinte forma: a seção 2 aborda os principais conceitos relacionados a gerência de identidades. A seção 3 apresenta a plataforma REALcloud, onde a arquitetura proposta está inserida. Na seção 4 é apresentado um estudo de caso no domínio da robótica em rede. Nas seções 5 e 6 apresenta-se os trabalhos relacionados, as conclusões e trabalhos futuros, respectivamente.

2. Conceitos sobre Gerência de Identidades

Nesta seção são apresentados os principais conceitos relacionados a gerência de identidades. Os conceitos de usuário, identidade e credenciais relacionam-se com o sujeito e sua identificação em um dado contexto. Os conceitos de autenticação e autorização relaciona-se com mecanismos para aplicar o controle de acesso. Federação refere-se a mecanismos para criação de parcerias com o objetivo de promover o compartilhamento de recursos. O SSO refere-se à maneira como o usuário de um serviço compartilhado pode se autenticar sem ter que utilizar inúmeras credenciais. A gerência de identidades trata então de oferecer um controle de acesso (autenticação e autorização) federado, oferecendo ao usuário uma experiência de SSO e ao administrador a provisão de contas (distribuídas entre diversas bases do domínio) e auditoria do sistema [Olden 2011].

2.1. Usuário, Identidade e Credenciais

Um usuário pode ser caracterizado como um cliente ou entidade (*host*/aplicação) que quer obter algum serviço de um provedor. O usuário pode ser uma pessoa utilizando um agente

de interface (por exemplo, navegador Web) ou um sistema computacional (por exemplo, *web services*). Um requisito necessário para que o usuário possa obter um recurso de um provedor é possuir uma identidade que o represente.

O conceito de identidade, segundo [Cao and Yang 2010], é difícil de precisar, uma vez que a definição de identidade está relacionada ao ambiente onde é empregada, a contextos semânticos e a casos de uso. Como uma definição mais geral, podemos dizer que uma identidade é uma representação de uma entidade em um contexto particular [El Maliki and Seigneur 2007]. Uma identidade consiste de identificadores e credenciais de usuários.

Uma credencial é um atestado de qualificação, competência ou autoridade, expedida a um indivíduo por terceiros com autoridade relevante ou competência para tal ato. Na computação, exemplos de credenciais incluem certificados digitais X.509 assinados por uma autoridade certificadora (CA), usuário e senha, *one time password* (OTP), *tokens*, entre outros.

2.2. Controle de Acesso (Autenticação e Autorização)

O processo de verificar a existência de uma identidade é chamado de autenticação. Para que tal processo se inicie é necessário que o usuário forneça uma credencial válida para o método de autenticação utilizado pela entidade autenticadora. De posse da credencial válida, a entidade autenticadora poderá verificar em sua base de identidades se há alguma identidade cuja credencial seja a fornecida pelo usuário. Caso a entidade autenticadora encontre alguma entrada referente a credencial, o usuário é autenticado. Com isso entende-se que o usuário provou ser quem ele informou ser.

Uma vez realizada a autenticação do usuário, a entidade autenticadora poderá verificar qual ou quais são as permissões de acesso ao recurso originalmente requisitado. A este processo de verificar as permissões de acesso a um determinado recurso chamamos de autorização.

2.3. Federação e SSO (Single Sign-On)

Uma federação respresenta um conjunto de organizações que cooperam entre si de acordo com regras de confiança pré-estabelecidas para a autenticação de usuários e compartilhamento de recursos [SWITCH 2011]. Uma solução de federação é considerada desejável quando organizações crescem com a aquisição de novos sites, para a manutenção distribuída de repositórios e para simplificar a autenticação e autorização de usuários a recursos e aplicações entre domínios parceiros [Ping Identity 2010]. As entidades a seguir são normalmente encontradas em uma federação:

- *Service Provider (SP)*: É a entidade mais próxima ao recurso do domínio em questão. É responsável por verificar a validade dos *cookies* de sessão e das permissões de acesso aos recursos por usuários autenticados. Como exemplo, o SP poderia ser um provedor, que oferece serviços de nuvem para seus clientes;
- *Identity Provider (IdP)*: É um provedor de serviços de identidades. Sua responsabilidade é de manter a base de dados de usuários do domínio e validar as credenciais de usuários. Como exemplo, pode ser uma empresa que gerencia contas para um grande número de usuários que precisam de acesso seguro a transações bancárias. Por meio dessa entidade os usuários fornecem as suas credenciais para acessarem os recursos federados;

- *Identity Provider proxy (IdPproxy)* ou WAYF (*Where Are You From*): É a entidade responsável por interrogar o usuário a respeito de seu domínio de origem. O domínio de origem é o local onde o usuário possui uma identidade em um IdP. Geralmente o IdPproxy apresenta uma lista de IdPs para o usuário selecionar o mais apropriado.

As duas primeiras entidades são geralmente encontradas em federações onde os participantes estão localizados em um mesmo domínio administrativo. Neste caso dizemos que a federação é intra-domínio e ocorre entre as diferentes aplicações do domínio. Em federações cujos elementos estão dispostos em domínios administrativos distintos, dizemos que a federação é entre domínios. Neste caso, e dependendo da arquitetura adotada, pode existir a última entidade acima.

2.4. Gerência de Identidades em Nuvem

É interessante notar que o uso de uma gerência de identidades depende do tipo de modelo de nuvem: para SaaS (*Software-as-a-Service*), apenas o controle de acesso às aplicações é necessário; para PaaS (*Platform-as-a-Service*) e IaaS (*Infrastructure-as-a-Service*), é necessário o controle de acesso integrado tanto das aplicações quanto do sistema de software e demais recursos (*software* e *hardware*) acessíveis na plataforma [Celesti et al. 2010].

O modelo tradicional de se manter uma base de dados compartilhada de identidades para aplicações Web não é adequado para ambientes de nuvem. Isso porque o usuário pode ter conteúdo e permissões de acesso compartilhados em diversos sites, e em diferentes aplicações Web. Em virtude disso, o mapeamento de identidades para usuários torna-se inviável em decorrência da alta replicação de dados.

Em razão disso, os modelos de controle de acesso centrado no usuário são mais adequados para ambientes de nuvem: as requisições aos SPs são tratadas de acordo com a identidade do usuário e suas credenciais. O controle de acesso centrado no usuário precisa conter informações que definam unicamente um usuário no domínio. Isso implica em manter um contexto de informação por usuário, de forma a prover SSO. Uma vez que o usuário pode ter múltiplas identidades, cada uma delas deve estar mapeada para o mesmo usuário, ou seja, uma identidade federada.

3. Plataforma REALcloud

A plataforma REALcloud suporta o acesso seguro e federado a máquinas virtuais (VMs) em uma nuvem híbrida possibilitando oferecimento de serviços de infraestrutura (IaaS) e de plataforma (PaaS). No caso de IaaS a plataforma oferece o acesso seguro a VMs configuradas com recursos computacionais de propósito geral, tais como sistemas operacionais, ambientes de desenvolvimento, servidores, etc. No caso de PaaS a plataforma REALcloud permite proteger os recursos da plataforma sendo oferecida como serviço. A plataforma REALcloud utiliza serviços de *firewall* para garantir que somente usuários autenticados e autorizados tenham acesso aos recursos sob controle da plataforma.

3.1. Padrão Arquitetural Proposto

A arquitetura proposta é uma arquitetura em camadas. Arquiteturas em camadas são interessantes quando uma das camadas possui um conjunto de serviços e estes podem ser

utilizados por uma camada de nível mais geral [Buschmann et al. 1996]. A Fig. 1 mostra a arquitetura em camadas proposta.

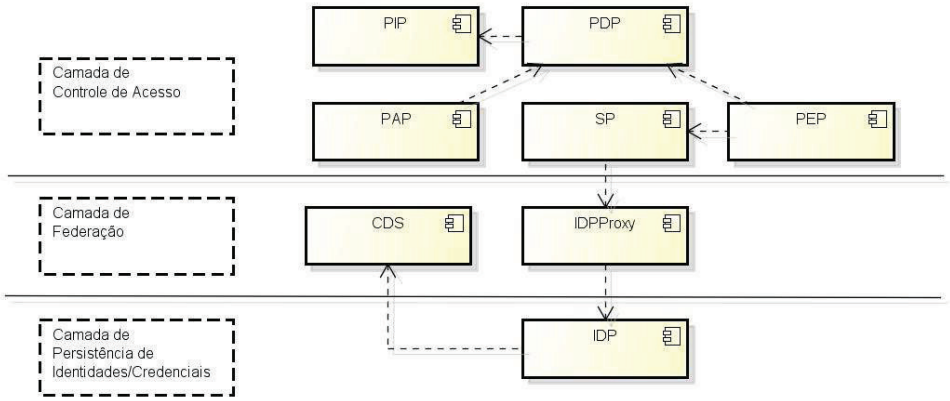

Fig. 1. Padrão arquitetural para gerência de identidades.

A arquitetura proposta está subdividida em 3 camadas a saber: *Controle de Acesso*, *Federação* e *Persistência de Identidades/Credenciais*. A camada de *Controle de Acesso* é responsável por proteger os recursos pertencentes aos provedores de serviços (SPs) e aplicar decisões referentes a autorização do usuário. Fazem parte desta camada os componentes de autorização PIP (*Policy Information Point*), PDP (*Policy Decision Point*), PAP (*Policy Administration Point*) e PEP (*Policy Enforcement Point*). O PEP é responsável por interceptar requisições a recursos do SP e aplicar a decisão de autorização vinda do PDP. O PDP é responsável por avaliar as políticas aplicáveis e devolver ao PEP uma decisão de autorização. O PAP é responsável pela criação das políticas. O PIP é responsável por obter informações/atributos adicionais e encaminhá-los ao PDP.

A camada de *Federação*, por sua vez, é responsável por prover serviços para localização de provedores de identidade (IdPs) e mecanismo de estabelecimento de sessão de federação. O componente que oferece o serviço de localização é o IdPproxy e o que trata do mecanismo de sessão de federação é o CDS (*Common Domain Services*).

A camada de *Persistência de Identidades/Credenciais* é responsável pela provisão (criação, deleção e edição) de identidades e credenciais. Esta camada é responsável pela manipulação das respectivas bases de dados. O componente IdP é responsável por estes serviços.

3.2. Arquitetura da Plataforma REALcloud

A plataforma REALcloud utiliza o padrão arquitetural apresentado na Fig. 1. A Fig. 2 apresenta uma visão geral da plataforma REALcloud. Nesta figura, cada nuvem possui os elementos necessários da arquitetura de gerência de identidades e cada elemento atua dentro do próprio domínio. O SP é responsável, juntamente com o elemento CAuth, por realizar o controle de acesso de recursos fornecidos em seu domínio. O CAuth que aparece em cada domínio, reúne os elementos PIP, PAP, PEP e PDP, com as funcionalidades descritas anteriormente.

O IdPproxy é responsável por oferecer um serviço de descoberta de provedores de identidades, e com isso, permitir que usuários de uma nuvem privada possa acessar

recursos (por exemplo, sua VM) da nuvem pública. O CDS é responsável por setar um *cookie* no navegador Web do usuário, representando que o mesmo já foi autenticado por um provedor de identidade participante da federação. O IdP, por possuir bases de dados de identidades, é responsável por fazer a autenticação dos usuários. É importante ressaltar que cada IdP possui uma base de dados de apenas usuários de seu domínio. Outro fato importante é que os elementos SP, IDPproxy e IDP estão distribuídos, em cada domínio, em VMs separadas para cada tipo de serviço. O elemento CAuth está distribuído na mesma VM junto ao SP e o elemento CDS está distribuído juntamente com a VM do IDPproxy.

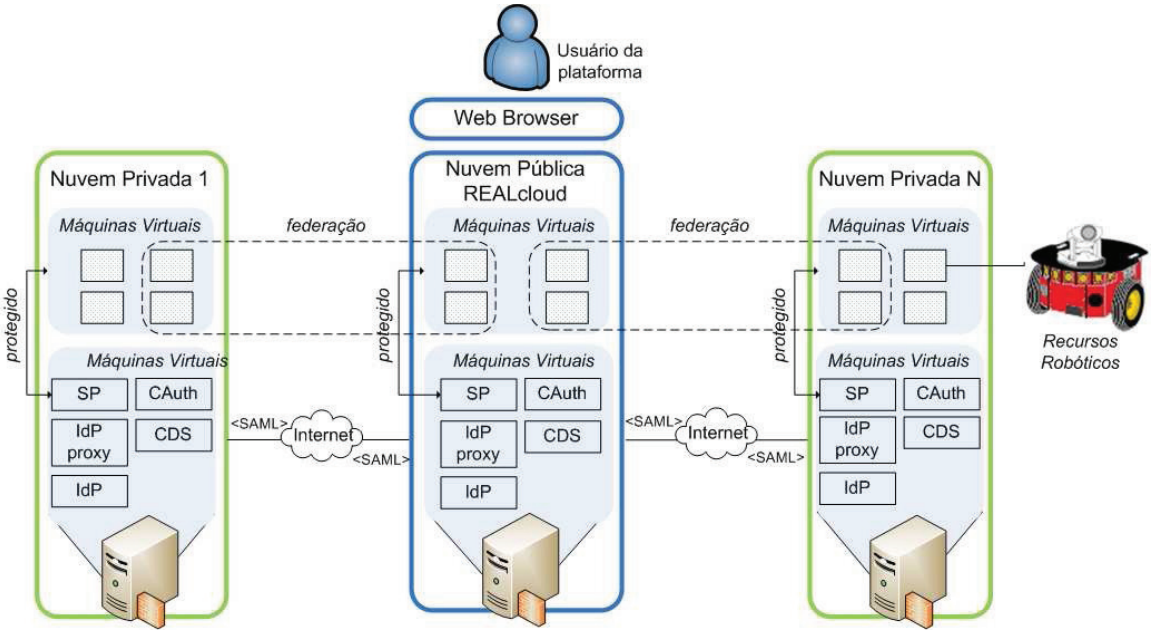

Fig. 2. Arquitetura da plataforma REALcloud.

A Fig. 3 mostra a distribuição dos componentes da plataforma REALcloud em máquinas virtuais.

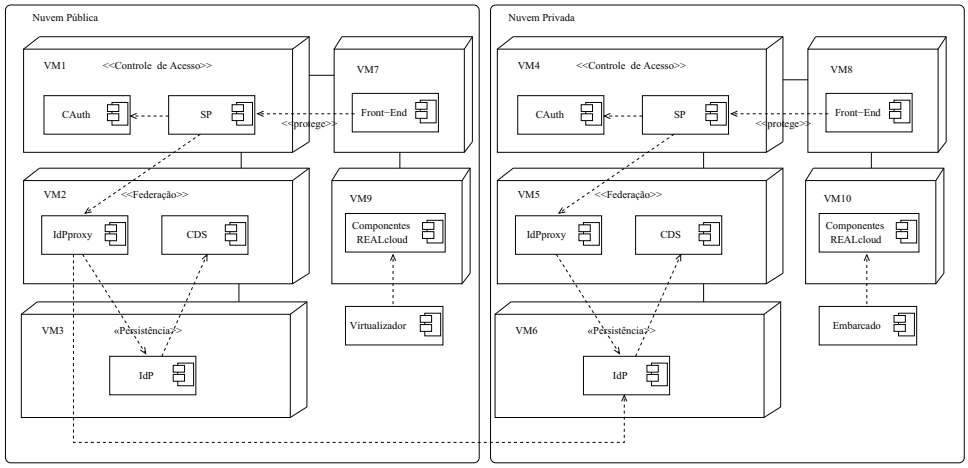

Fig. 3. Distribuição da arquitetura em camadas na plataforma REALcloud.

A distribuição dos componentes da arquitetura na nuvem pública ocorre com os

componentes *SP*, *IdP* e *IdPproxy* sendo distribuídos em máquinas virtuais. Cada máquina realiza uma camada da arquitetura proposta. Com isto obtemos uma forma não-invasiva de proteção dos recursos existentes na nuvem. O componente *Front-End* implementa o serviço de gerência de VMs. O componente *Virtualizador* implementa a tecnologia de virtualização utilizada pelos componentes da plataforma REALcloud. Na nuvem privada, o componente *Front-End2* exibe a interface para a interação com os recursos robóticos. O componente *Embarcado* implementa microservidores HTTP para interação remota na rede interna da nuvem. Os *componentes REALabs* interagem com os recursos robóticos por meio do componente *Embarcado*. Para que um usuário consiga se identificar na plataforma REALcloud, o componente *IdPproxy* da nuvem pública contém informações referentes ao componente IdP da nuvem privada.

3.3. Aspectos da Implementação

A implementação da arquitetura utilizou como base a infraestrutura oferecida pelo *middleware* OpenAM. O OpenAM é responsável por prover serviços de controle de acesso (autenticação e autorização) federado, SSO e uma simples gerência de usuários. Os serviços de controle de acesso, verificação de *tokens*, *logging* e provisão de identidades, oferecidos pelo OpenAM, podem ser acessados tanto via serviços Web, via HTTP, ou por meio de um agente, podendo ser visto como um modelo SaaS de serviços de controle de acesso e identidade [ForgeRock 2010].

Para realizar a autenticação dos usuários, o OpenAM também oferece a possibilidade de instalação de um agente. O agente funciona como um filtro e pode ser instalado em diversos servidores de aplicação (Apache, Tomcat, Glassfish, etc.). O agente atua como um PEP e a filtragem ocorre na requisição de um recurso hospedado no servidor. Na nuvem pública foi utilizado um agente no Tomcat para interceptar requisições a página de gerência de VMs do REALcloud. Na nuvem privada, onde está localizado o REALabs, foi instalado um agente no Apache para proteger páginas de gerência do laboratório (reserva de horário, provisão de experimentos e etc.). Neste servidor Apache também está instalado o módulo *mod_proxy*. Com este módulo instalado, obtemos uma maior proteção na utilização dos recursos uma vez que podemos atribuir endereços de rede privados aos recursos. Desta forma, uma máquina virtual localizada na núvem pública, não conseguirá interagir diretamente com estes recursos. como segurança adicional, neste mesmo *proxy* e nos IdPs de cada domínio, foi congurado uma conexão HTTPS.

O estabelecimento do círculo de confiança (CoT) da federação se deu através da configuração de um perfil (SP, IdP ou IdPproxy) em cada OpenAM. Para tal, foi necessário a criação de um metadata no padrão SAMLv2. Além de conter informações sobre o perfil, este metadata, utilizado na comunicação entre as entidades, possui um certificado X.509 assinado por uma entidade certificadora. Os metadatas criados nas entidades IdP e SP foram exportados para a entidade IdPproxy. Na entidade SP, foi importado o metadata referente ao perfil IdP da entidade IdPproxy. Na entidade IdP, foi importado o metadata referente ao perfil SP da entidade IdPproxy. Assim, o IdPproxy atuará, sob o ponto de vista do SP, como um IdP e do ponto de vista de um IdP, como um SP. Isto foi importante para permitir a agregação de múltiplos IdPs (de domínios diferentes) para fazer parte da federação.

O acesso às VMs é realizado por meio de um IP público e porta de conexão (*port forwarding*). Para que as VMs tenham acesso à Internet, é utilizado NAT (*Network*

Address Translation) para prover a conversão de endereços IP privados em endereços públicos. A plataforma REALcloud utiliza o módulo *iptables* disponível nas plataformas Linux para prover NAT e *port forwarding*.

4. Estudo de Caso

A platforma REALcloud está sendo utilizada no suporte à realização de experimentos robóticos em nuvens computacionais. Neste caso, a plataforma REALcloud oferece a plataforma REALabs como serviço. A plataforma REALabs, por sua vez, é uma plataforma de robótica onde os seus recursos físicos podem ser controlados via rede. Dentre os componentes desta plataforma estão interfaces Web de gerência para a configuração de experimentos e reserva de horário. Os recursos físicos desta plataforma são câmeras, sonares, lasers, garra robótica, dentre outros. Estes recursos podem ser acoplados a um robô móvel para a realização de experimentos, por exemplo, experimentos de localização e navegação.

O *middleware* OpenAM foi estendido com a implementação de um mecanismo de autorização básica para o acesso aos recursos robóticos. Este mecanismo consiste em invocar um *servlet* Java de autorização após uma autenticação bem-sucedida. Este *servlet* insere um conjunto de regras *iptables* para liberação do uso de recursos da rede interna da nuvem. As regras se referem ao IP de origem da VM do usuário e ao IP de destino do recurso. Ao final de cada experimento, ou no final de uma sessão de acesso, o serviço de controle de acesso localizado na nuvem privada atua e bloqueia o acesso da VM à rede de recursos. No entanto, o usuário ainda tem o acesso à sua VM.

4.1. Acesso aos Recursos

Para validar a plataforma REALcloud foi realizado um estudo de caso que consiste em utilizar a plataforma REALcloud para realizar um experimento robótico de *tracking* visual. Este experimento tem como objetivo fazer um robô perseguir outro robô utilizando sua câmera de bordo. O primeiro passo é o usuário fazer um agendamento de horário. Para tal o usuário acessa o serviço de reserva de horários da plataforma REALabs e seleciona um horário disponível. É importante ressaltar que para o usuário acessar o serviço de reservas, antes será realizado um redirecionamento pelo PEP para que ele se autentique. A Fig. 4 apresenta uma visão geral do processo de estabelecimento da autenticação federada na plataforma REALcloud e autorização para realização do experimento. Por razões de simplificação, neste diagrama de sequência, os componentes PEP, PIP, PAP e PDP se encontram dentro do componente SP.

No passo 1, a partir do navegador Web (*browser*) o usuário solicita o acesso ao serviço de reserva de horários. Neste momento, o PEP localizado no servidor de aplicação onde o serviço está disponível, intercepta a requisição, verifica que o usuário não está autenticado (isto é, não possui um *cookie*) e redireciona ao SP do OpenAM. O redirecionamento é uma mensagem do tipo HTTP 302 *Redirect*, informando que o recurso solicitado moveu-se para outra URL. O SP inicia o processo de autenticação federada, criando uma requisição SAML (*AuthnRequest*). Esta requisição é encapsulada na resposta de redirecionamento, que contém o endereço do IdPproxy para a autenticação. No passo 2, o navegador Web do usuário é redirecionado para o IdPproxy.

Neste passo, o IdPproxy apresenta uma lista para que o usuário escolha em qual domínio deseja ser autenticado, conforme apresentado pela Fig. 5, e também, uma nova

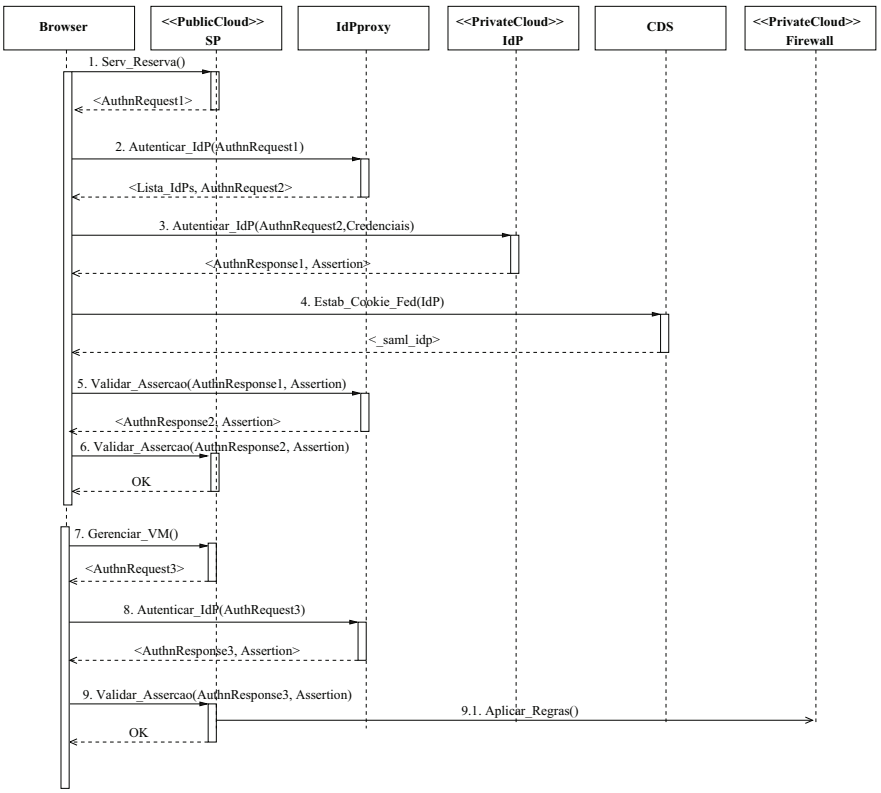

Fig. 4. Estabelecimento da autenticação federada e autorização na nuvem.

requisição SAML (*AuthnRequest2*) é gerada pelo IdPproxy. O usuário escolhe se autenticar em um IdP da nuvem privada e o navegador Web é, então, redirecionado para a autenticação neste IdP escolhido. O passo 3 demonstra o usuário sendo redirecionado para o IdP escolhido, bem como o navegador Web fornecendo a requisição SAML e o usuário fornecendo suas credenciais (no caso, usuário e senha). De posse das credenciais, o IdP verifica se ela é valida e após isto o usuário está autenticado. O IdP irá então emitir uma resposta SAML à requisição feita e irá redirecionar o navegador Web para o CDS (passo 4) a fim de inserir o *cookie* de federação (_saml_idp) no navegador Web do usuário. É importante ressaltar que este passo é importante porque garante que nos próximos acessos a URLs de outros domínios federados, não será necessário informar novamente suas credenciais. Neste passo também contém uma informação referente ao redirecionamento para o IdPproxy prévio e a asserção expedida pelo IdP.

No passo 5, o navegador Web é redirecionado para o IdPproxy, onde é informado a asserção SAML prévia. O IdPproxy valida esta asserção e gera uma nova resposta (*AuthnResponse2*) que deverá ser enviada ao SP. Neste passo uma informação de redirecionamento ao SP é adicionada. Assim, no passo 6, o navegador Web é redirecionado para o SP, com a asserção contida em *AuthnResponse2* e o *cookie* de federação. O PEP do SP irá novamente interceptar esta requisição, porém, desta vez, ele verificará a presença da asserção e do *cookie* de federação. O PEP verificará junto ao SP que a asserção é de uma entidade pertencente ao CoT e irá, por sua vez, assumir que o usuário está autenticado. Com isso, será enviado ao navegador Web do usuário a página do serviço de reserva. É importante ressaltar que esta autenticação na plataforma é federada e realiza o SSO entre todos os provedores de serviço existentes em todas as nuvens.

Após o passo 6, o usuário deverá inicializar a sua máquina virtual, que está localizada junto a rede física do robô. Para inicializar a sua máquina virtual, o usuário deverá acessar o serviço de gerência de VMs. O passo 7 se inicia com a tentiva do usuário em acessar este serviço. Novamente o PEP do SP irá interceptar esta requisição e irá encaminhar ao SP para gerar uma requisição SAML (*AuthnRequest3*), juntamente com um redirecionamento pra o IdPproxy conhecido por este SP. No passo 8 o navegador Web do usuário envia ao IdPproxy a requisição SAML prévia e este verificará a presença do *cookie* de federação contendo um valor referente a um IdP pertencente ao CoT conhecido.

Assim, o IdPproxy irá expedir uma reposta SAML (*AuthnResponse3*) contendo uma asserção e enviará como resposta ao navegador Web do usuário, juntamente com uma informação de redirecionamento ao SP detentor da página de gerência de VMs. Esta resposta SAML será enviada, no passo 9, ao SP em questão. O PEP do SP irá novamente interceptar esta requisição, porém, desta vez, ele verificará a presença da asserção e do *cookie* de federação. O PEP verificará junto ao SP que a asserção é de uma entidade pertencente ao CoT e irá, por sua vez, assumir que o usuário está autenticado. Com isso, será enviado pelo SP uma mensagem ao *firewall* da nuvem privada onde os recursos em questão estão, contendo regras do *iptables* para liberar o uso desta rede privada ao usuário. Na sequência, será enviado ao navegador Web do usuário a página do serviço de gerência de VMs. Por fim, o usuário poderá realizar seus experimentos a partir de sua máquina virtual.

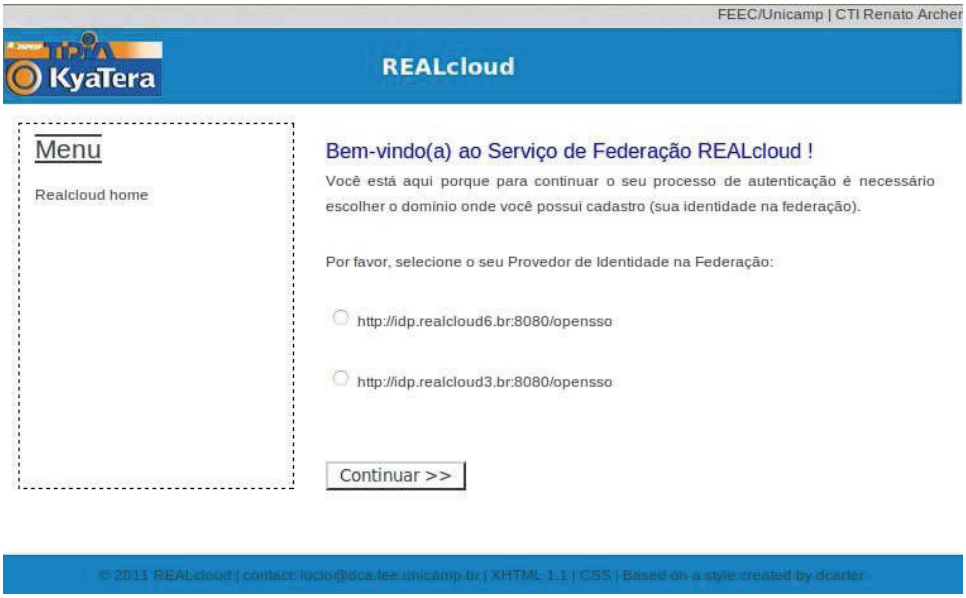

Fig. 5. Serviço de escolha de IdPs na plataforma REALcloud.

4.2. Execução do Experimento

O experimento de *tracking* visual utiliza um controlador baseado em regras de decisão *fuzzy*. O controlador utiliza informações extraídas da câmera fixada em um dos robôs. O objetivo do controlador é manter uma distância constante entre os robôs. Para isto, um algoritmo de visão computacional extrai as entradas a serem utilizado pelo controlador da imagem. Estas entradas são o centróide de uma área de interesse e a porcentagem que esta área de interesse ocupa na imagem.

No algoritmo de visão, a área de interesse corresponde aos pixels que possuem a cor do robô que se deseja perseguir. Para o robô utilizado, esta cor corresponde ao vermelho. Para isolar a cor de interesse na imagem, primeiramente o algoritmo de visão emprega técnicas de isolamento de cor no sistema de cores HSV. Em seguida, utiliza operadores morfológicos largamente utilizados na literatura como os de abertura e fechamento. O operador de abertura possui uma estrutura retangular de 4×2 pixels e o de fechamento, um disco de raio igual a 5 pixels. Desta forma é possível determinar o centróide e a porcentagem ocupada pela área de interesse, requisitos do controlador.

O controlador *fuzzy*, por sua vez, possui duplo objetivo. O primeiro é o de centralizar o centróide na imagem, fato que implica em alinhar o robô com o objeto perseguido, neste caso, outro robô. O segundo é o de manter a porcentagem da área ocupada pela cor do robô constante na imagem. Este segundo objetivo implica que a distância entre os robôs permanecerá constante. Com duas entradas, centróide e porcentagem de área ocupada, e duas saídas, velocidades dos motores esquerdo e direito, este controlador emprega o método de inferência de Mamdani, com operadores MAX-MIN e centróide como operador de defuzzyficação.

A título de comparação, o mesmo experimento foi realizado utilizando um computador conectado na rede Internet executando o experimento de *tracking* visual. O atraso da rede causa uma degradação de desempenho do controle, o que faz com que a trajetória do robô que persegue o alvo seja irregular. Isto é ilustrado na Fig. 6(a) onde a frequência de atuação do controle foi de 4,55 atuações por segundo com desvio padrão de 1,301. Nesta figura, a linha pontilhada é a trajetória do robô alvo e a linha cheia a trajetória do robô perseguidor.

A Fig. 6(b) ilustra o mesmo experimento executado em uma máquina virtual da plataforma REALcloud. Neste caso, o atraso é bastante reduzido quando comparado com o atraso experimentado na Internet. Como consequência, o robô desenvolve uma trajetória homogênea, praticamente idêntica à trajetória do robô alvo. Neste caso, a frequência média do controle foi de 12,49 atuações por segundo com desvio padrão de 1,007.

Além de propiciar acesso seguro, a plataforma REALcloud propicia que a execução dos experimentos ocorra em servidores localizados a apenas um *hop* da rede (privada) onde os recursos se encontram.

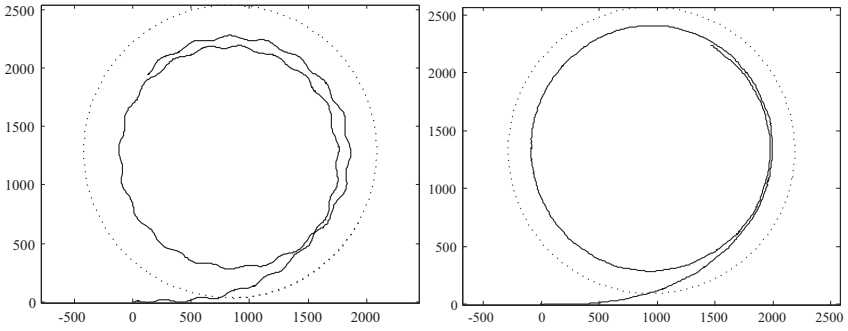

Fig. 6. Experimentos realizados: (a) execução via Internet, (b) execução na plataforma REALcloud.

5. Trabalhos Relacionados

O trabalho descrito em [Emig et al. 2007], apresenta uma arquitetura em camadas semelhante à arquitetura proposta neste artigo. Em relação à arquitetura proposta, utilizamos o perfil de SSO chamado *SP-Initiated SSO* ao invés do *IdP-Initiated SSO*, proposto pelo outro autor. O modelo *SP-Initiated* é mais interessante do ponto de vista da experiência do usuário, dado que a solicitação de credenciais ocorrerá após a tentativa de acesso a um recurso. Este modelo de SSO foi definido com a especificação do SAML versão 2.0 [OASIS 2008], que atualizou a antiga versão 1.0. Outra questão interessante foi que os autores do trabalho não dividiram sua arquitetura em componentes responsáveis por prover os recursos (SP) e por fazer a autenticação (IdP). Assim como descrito em [Celesti et al. 2010], acreditamos que este isolamento na arquitetura é importante, especialmente para ambientes em nuvem, pois permite realizar a autenticação em um domínio diferente do domínio do recurso solicitado.

Os problemas referentes à gerência de identidade surgiram em meados da década de 80, destacando-se o sistema Rec. X.500 para serviços de diretórios do tipo *Directory Access Protocol*(DAP). Na década de 90, o IETF padronizou o *Light Directory Access Protocol* (LDAP), muito utilizado nos primeiros navegadores Web, como o Netscape Navigator. Na época, a Microsoft propôs um padrão similar conhecido como *Active Directory*, e serviços como o *Passport* [El Maliki and Seigneur 2007]. Atualmente destacam-se paradigmas orientados ao usuário, o que é conhecido como *Identity Management (IDM) 2.0*. A seguir citamos algumas propostas nesta linha.

Para a representação universal e abstrata de identificadores, conveniente para sistemas com SSO, a OASIS propõe o uso do protocolo XRI [OASIS 2011]. Esse protocolo pode ser utilizado para identificar recursos independente do domínio, sejam eles *hosts*, *softwares* ou pessoas. O *Identity Web Services Framework* (ID-WSF) da Liberty Alliance [Liberty Alliance 2011] surgiu como um *framework* para estabelecer o uso de padrões abertos para a identidades federadas. Além de adotar o modelo de CoT para estabelecer o SSO entre SPs e IdPs, o *framework* se preocupou com a segurança das mensagens trafegadas entre os domínios, sugerindo o uso de especificações como o WS-Security, WS-Policy, WS-Trust e WS-Federation, juntamente com asserções SAML. O *framework* também especifica como mensagens de autorização e autenticação em SOAP (*Simple Object Access Protocol*) podem trafegar em domínios federados com integridade e confidencialidade.

O Shibboleth [Internet2 2011] foi uma das primeiras iniciativas de código aberto para a integração de recursos Web em universidades. Ele é um *middleware* para Web SSO federado desenvolvido pelos grupos Internet2 e *Middleware Architecture Committee for Education* (MACE). Shibboleth utiliza asserções SAML e os conceitos de provedores de serviços (SP), provedores de identidades (IdP) e serviços de listagem de domínios parceiros (*Discovery Service* ou WAYF).

O projeto Higgins [Eclipse Project 2011] é um *framework* de código aberto para a integração de identidades, perfis e informação distribuída. Um *Active Client* integrado ao navegador Web do usuário mantém cartões com informações de contexto e credenciais. Nesses cartões (I-Cards) há informação suficiente para acessar Web sites sem a necessidade de informar a senha, ou de preencher formulários para cadastro. Plugins para a IDE Eclipse também estão disponíveis para o desenvolvimento de aplicações Web integradas.

Outros trabalhos na literatura [El Maliki and Seigneur 2007] descrevem que a sociedade da Internet tende a se preocupar cada vez mais com a privacidade dos usuários, e com a administração de suas identidades digitais. Isso implica na necessidade de plataformas escaláveis para prover gerência de identidade. Este fato justifica a necessidade em nosso trabalho de uma solução capaz de tratar questões como a integração de diferentes mecanismos de autenticação, integração de diferentes tecnologias de bases de dados e a possibilidade de oferecer um ambiente de gerência de identidades como serviço, oferecendo às diversas aplicações interfaces de acesso aos serviços de identidade (por exemplo, *web services*, interfaces HTTP, entre outros). Com o uso do *middleware* OpenAM, conseguimos equacionar adequadamente estas questões.

Ressaltamos também, que a arquitetura proposta é independente de tecnologia de *middlewares* para controle de acesso federado. Por estar em camadas, a arquitetura proposta prevê interoperabilidade entre seus componentes, utilizando diferentes *middlewares*, desde que eles tenham protocolos compatíveis (por exemplo, SAMLv2). Como exemplo, poderíamos ter o componente SP como uma instância do SP Shibboleth, comunicando-se com um IDP, como uma instância do IDP OpenAM. Também, os componentes da arquitetura podem ser substituídos sem causar prejuízos na disponibilidade do serviço de identidade da nuvem, pois estes estão instanciados em VMs.

6. Conclusões e Trabalhos Futuros

Neste artigo apresentamos uma arquitetura para realizar a gerência de identidades em nuvens híbridas. Esta arquitetura permite que outras nuvens privadas se associem à nuvem pública da plataforma REALcloud. A implementação desta arquitetura utilizou como base a infraestrutura oferecida pelo *middleware* OpenAM. Na distribuição dos componentes da arquitetura foram configuradas máquinas virtuais com este *middleware* e configuradas para prover a gerência de identidades na plataforma. O uso da virtualização é uma alternativa para prover maior disponibilidade dos serviços de proteção de recursos na nuvem, além de simplificar a implantação de sistemas federados.

O *middleware* OpenAM foi estendido para comportar serviços de autorização à rede interna de recursos. Para isto foi desenvolvido um *servlet* que insere regras de *firewall* na tabela de roteamento da nuvem privada da plataforma. Com relação a federação, do ponto de vista do administrador, sistemas federados simplificam a administração de contas de usuários dos diversos domínios existentes. Do ponto de vista do usuário, domínios federados reduzem a quantidade de autenticações necessárias para interagir com os serviços ofertados. Um sistema para gerência de identidade, portanto, deve integrar o uso de diversas formas de autenticação e autorização entre as diversas bases de dados de usuários, distribuídas em diversos domínios federados e oferecer ao usuário uma experiência de SSO. A plataforma REALcloud implementa um modelo virtualizado em camadas para a gerência de identidades em nuvems híbradas. Apresentamos um estudo de caso que demonstra a aplicação da proposta para realização de experimentos de robótica em rede.

Pretendemos, como trabalho futuro, estender o mecanismo de autorização para contemplar políticas de acesso de granularidade fina (*entitlements*) e aplicar no acesso de todos os recursos existentes na plataforma REALcloud. Um caminho a ser seguido seria no uso de ontologias para descrever os recursos da plataforma e suas possíveis composições. E, a partir destas ontologias, aplicar as políticas de autorização.

Referências

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M., and Stal, M. (1996). *Pattern-Oriented Software Architecture Volume 1: A System of Patterns*. Wiley.

Cao, Y. and Yang, L. (2010). A Survey of Identity Management Technology. *IEEE International Conference on Information Theory and Information Security (ICITIS)*.

Celesti, A., Tusa, F., Villari, M., and Puliafito, A. (2010). How to Enhance Cloud Architectures to Enable Cross-Federation. *IEEE 3rd International Conference on Cloud Computing*.

Eclipse Project (2011). Higgins - Open Source Identity Framework. Disponível em: http://www.eclipse.org/higgins.

El Maliki, T. and Seigneur, J.-M. (2007). A Survey of User-centric Identity Management Technologies. In *Proceedings of the The International Conference on Emerging Security Information, Systems, and Technologies*, SECUREWARE '07, pages 12–17, Washington, DC, USA. IEEE Computer Society.

Emig, C., Brandt, F., Kreuzer, S., and Abeck, S. (2007). *Identity as a Service - Towards a Service-Oriented Identity Management Architecture*. Springer-Verlag.

ForgeRock (2010). OpenAM. Disponível em: http://www.forgerock.com/openam.html.

Internet2 (2011). Shibboleth - A Project of the Internet2 Middleware Initiative. Disponível em: http://shibboleth.internet2.edu.

Liberty Alliance (2011). Disponível em: http://projectliberty.org/resource_center/specifications.

OASIS (2008). Security Assertion Markup Language (SAML) V2.0 Technical Overview. Technical report. Disponível em: http://docs.oasis-open.org/security/saml/v2.0.

OASIS (2011). Disponível em: http://www.oasis-open.org/committees/xri/.

Olden, E. (2011). Architecting a Cloud-Scale Identity Fabric. volume 44, pages 52–59. Journal Computer - IEEE Computer Society.

Ping Identity (2010). About Identity Federation and SSO. Disponível em: http://pingidentity.com.

Rocha, L. A., Olivi, L., Paolieri, F., Feliciano, G., Souza, R. S., Pinho, F., Teixeira, F., Rodrigues, D., Guimaraes, E., and Cardozo, E. (2011). *Computer Communications and Networks - Advances in Educational Robotics in Cloud with Qualitative Scheduling in Workflows*. Springer, University of Derby.

SWITCH (2011). SWITCH - Serving Swiss Universities. Disponível em: http://www.switch.ch.